DRAFT

CLCS System Build Procedure

1. Introduction

This document describes the CLCS System Build Procedure as it applies to software files. For the purposes of this document, a System Build currently refers to any SCID system build. System Builds are an important activity that culminate during and after the development of the delivered baselines (i.e. Redstone, Thor, Atlas, etc.).

2. System Build Procedure Scripts

The System Build Procedure relies on several scripts that were written specifically for an SCID build. They are listed below:

a. defsub
b. prepare_build
c. fetch_All
d. check_fetch
e. build.redstone
f. defscid
g. prepare_scid

3. System Build Login Requirements

In order for a user to perform a successful System Build, they must have access to a UNIX workstation. The user must also log on as scidbld. It is important to remember to log on to the proper server for the specific build you will be performing. Listed below are the servers you should be logged onto for a System Build:

	Server	Type of Build	Subsystem Load baseline
a.	razor	Origin	ccp-ddp.origin.redsone...
b.	sde1ddp1	Origin	ccp-ddp.origin.redstone...
c.	enterprise	Gateways/Sim	gw.csg.redstone...
d.	local machine	O2	hci.o2.redstone...

4. System Build Procedure

Listed below are the actual steps that would need to be executed to ensure a successful System Build.

a. Log onto a server as scidbld by entering the following at the command prompt: ***rlogin <server name> -l scidbld***

Tip: If you are logging onto the Gateway server (enterprise) do not use the server name. Instead, use the server's IP address: 163.206.64.33.

You will then be prompted for your password.

b. At the command prompt, enter ***defsub***. Enter the name of the subsystem you will be defining. *Ex: hci.o2.redstone*

c. You will be asked **Enter new version number:**. Select the enter key to increment the version number. Script will tell you the new version number.

d. The defsub script will compare each baseline in the previous version of the Subsystem Baseline load with the version in the repository. Whenever versions mismatch, the defsub script will say:
VERSIONS DIFFER for: <<subsystem load baseline>>. Select the default to keep the newest version.

e. If the question asks: **in cm, but NOT IN BL: <subsystem load baseline> Add this?:**, select the default (no) which will not add the baseline.

f. You also have the option of adding or removing a baseline. Selecting default to each question will neither add or remove baselines.

g. A summary of any baseline changes to the Subsystem load baseline will be displayed before the defsub script exits, saving any changes.

h. Type in the command: ***prepare_build <baseline> <version number>***.
Ex: prepare_build hci.o2.redstone 1.11
Tip: Make sure you put a space between the baseline and the version.

i. Change default to the new build directory that was just created. You will see several new files created, in particular fetch_All and build.redstone.

j. Type ***fetch_All*** at the command prompt. This will fetch the proper files for the build.
Tip: Perform the fetches on an Origin server (razor or sde1ddp1). The fetch will finish much faster due to the multi-processing capabilities of an Origin.
Tip: Perform the Gateway fetches on the enterprise server. The directory structure is set up differently on these servers, so you may receive an error message when you attempt to run a Gateway fetch on a non-gateway server.

k. To check on the status of the fetches, open up another UNIX shell window and type in the command: ***ps -ef | grep fetch*** or use ***top***. When the fetches are completed, all PIDs will have terminated.

l. After the fetches are complete, change default to the logs directory by typing in ***logs*** at the command prompt. Next, change default to the new build directory that was created. Type in ***check_fetch*** to check the log file for fetch errors.

m. If there are no errors, type in ***working*** to get you back to the working directory, Then ***cd*** to the build directory you are working on.

n. Execute the command: ***build.redstone <platform> <scid name>.<version #>***.

Ex: build.redstone csg gw.csg.redstone.1.3
build.redstone sim sim.gse.redstone.1.1
build.redstone o2 hci.o2.redstone.1.12
build.redstone origin ccp-ddp.origin.redstone.1.5

Tip: Make sure you are executing the build on the correct server. See the System Build Login Requirements section.

o. Gain access to another UNIX shell window and check the logs directory periodically for exec or lib compile errors. Use the command ***grep "Error code" *lib*** then ***grep "Error code" *exec*.***

p. If the build is clean, type in the command ***defscid***. Enter the name of the SCID you created.

Ex: defscid scid_ops.redstone

The script will prompt you to for an SCID version number, and whether to ***include***, ***exclude***, or ***update*** the subsystem load baselines within the SCID you just created.

q. Next, enter the command ***prepare_scid***. This script will create all the baseline soft links in the scid_ops directory. You will be asked to enter an SCID name. Enter the name of the SCID you entered in step p.

r. If you only need a subsystem load baseline build, then follow the steps above and create soft links and a directory using the following format: ***scid_<csc>_<test level>.<delivery>.<version>***

Ex: scid_gse_cit.redstone.1.1
scid_sct_uit.redstone.1.1

m. After the build is complete, you can create the Build Report. Change default to /net/user8/CLCS/Build_area/SBR/SCIDs/Formal/Logs. At the command prompt, enter the command:
build_report scid_ops.redstone.<version #> > build_report.scid_ops. redstone. <version #>.

5. **Creating SCID Baseline Using Razor**

After the build report is done, you must create the SCID baseline in Razor. Storing the SCIDs in the repository permits the fetching of all files, baselines, and scripts that were used at the time of the System Build. The procedures are listed below.

a. Bring up the Razor client ***Threads***. Select **File**, then **New Thread**.

b. Be sure to include the following in the SCID baseline you are creating:

Folder	**Files and/or Subsystems**
SCIDs	scid_ops.redstone
Subsystems	all subsystems created during build
Tools/Build	build.redstone
Misc	all files
PostBuild	all files
PreBuild	all files

c. Save the SCID as ***scid_ops.redstone.<version #>***. The version number should coincide with the version number that was entered in step p.

6. **Rebooting Razor Database**

You may encounter problems with Razor during a system build. If the Razor database should crash during a build, follow these instructions:

a. Log onto CLCS17.

b. Enter clcscm (will set up your RAZOR_UNIVERSE environment).

c. Type in **razor up** at the command prompt.